PipeRench : Power and Performance Evaluation of a Programmable Pipelined Datapath

Benjamin A. Levine and Herman H. Schmit

Dept. of Electrical and Computer Engineering

Carnegie Mellon University

Pittsburgh, PA USA 15213

blevine@cmu.edu, herman@ece.cmu.edu

Electrical & Computer ENGINEERING

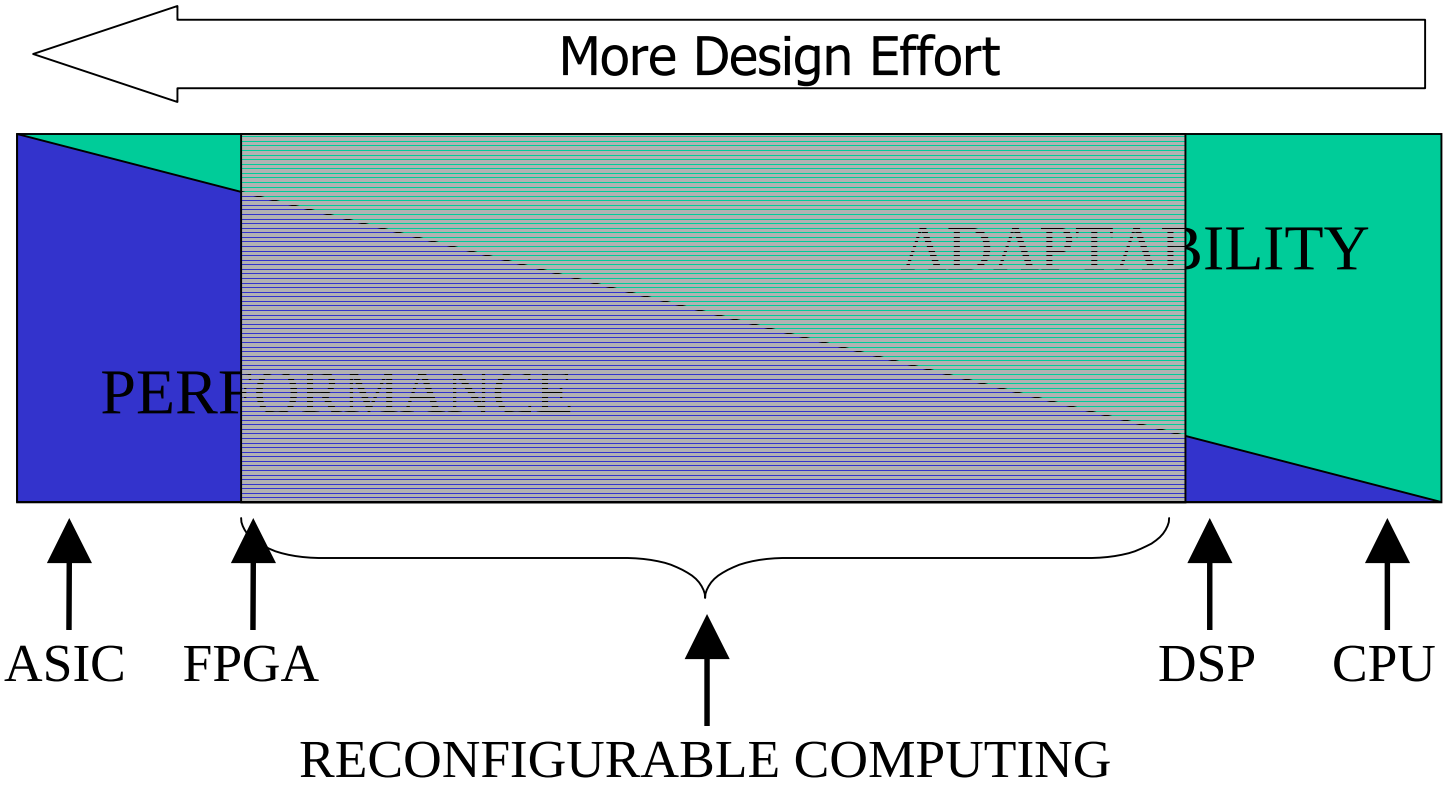
PipeRench = **Reconfigurable Computing Device**

Reconfigurable Computing Device = A Computing Device which can be reconfigured for each different application that it runs, by changing the functionality of its hardware and the way that its hardware is connected.

PipeRench **was developed by students and faculty at Carnegie Mellon.**

Electrical & Computer ENGINEERING

Why Reconfigurable Computing?



Why Reconfigurable Computing?

We want performance **and** adaptability:

- Performance of an ASIC

  Implement application as custom datapath to:
  - Increase parallelism.
  - Decrease memory traffic (through locality).
  - Increase performance.
  - Use less power.

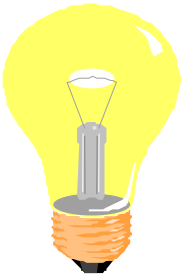
- Adaptability of a CPU.

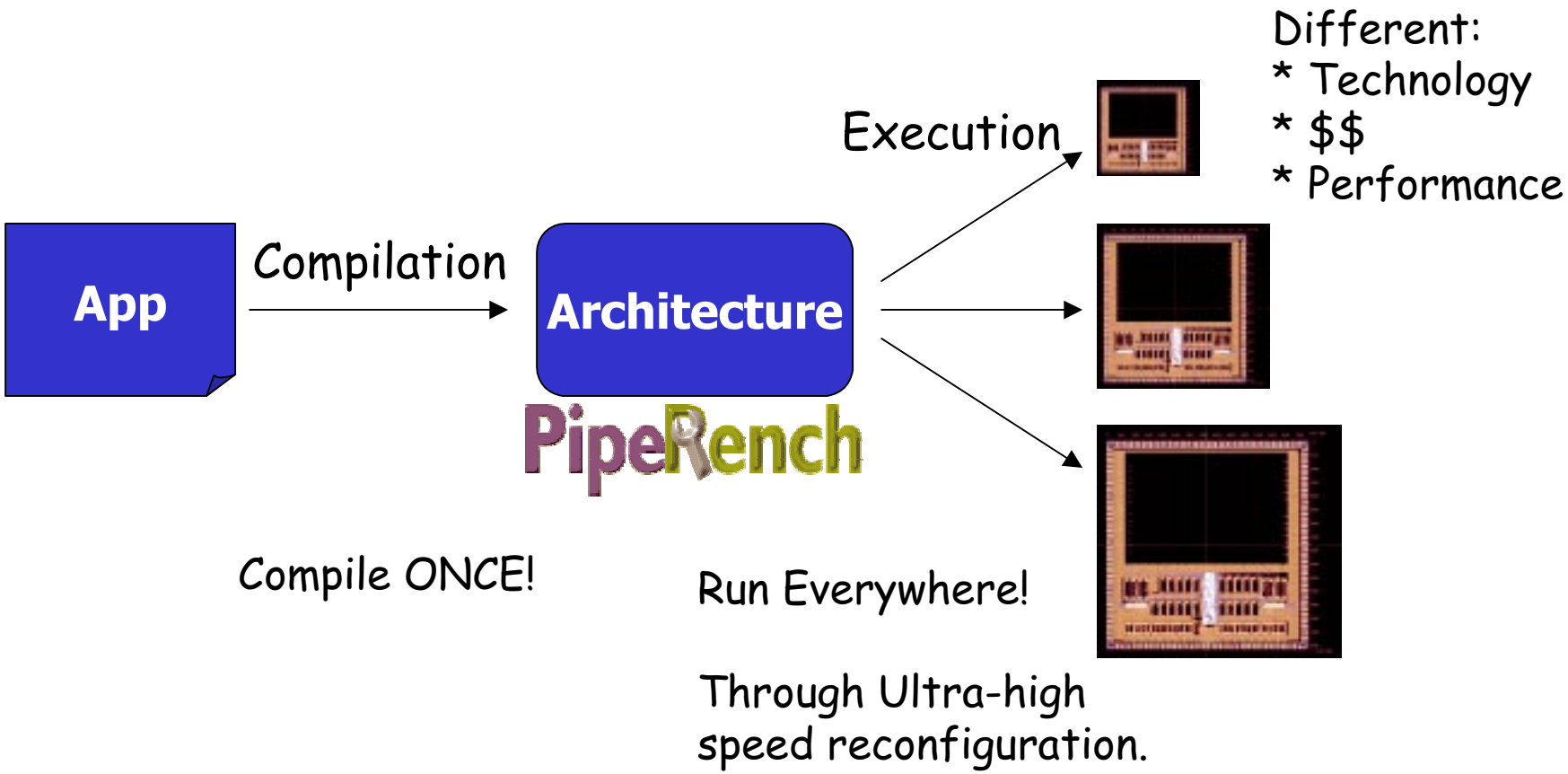
  Completely reprogram as needed for new applications.

Why NOT Reconfigurable Computing?

- FPGA design is more like HW than SW
  - No real C to FPGA yet, so must use HDL
- FPGA configuration is fixed to one FPGA
  - Must redesign to gain performance on larger FPGAs
  - Can't use design on FPGA with fewer resources.
  - Compares poorly to SW for microprocessors:
    - No portability
    - No scalability

Solution:
Use A Virtual Architecture



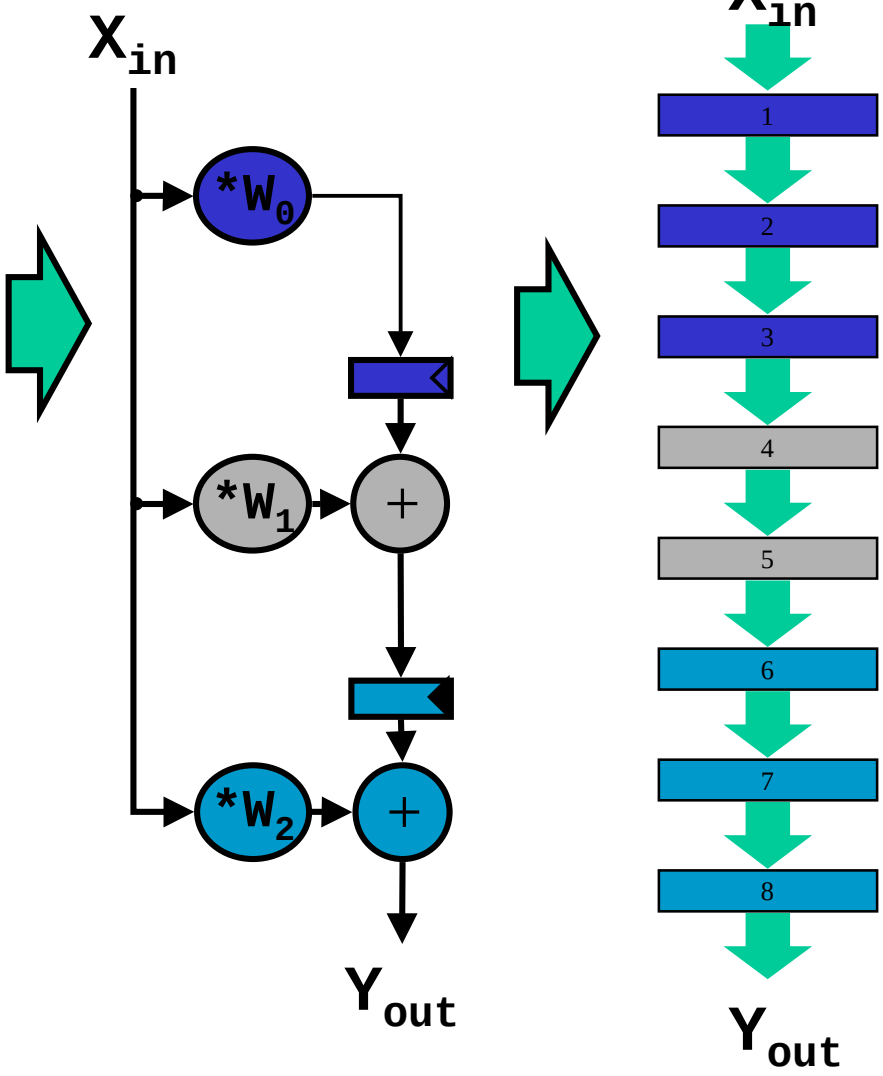


Virtual Architecture

- Compile to virtual machine
  - Makes compilation easier
  - Compile from high-level language (DIL)
  - Binaries decoupled from specific hardware
  - Scalable / Re-usable
- Restrict the model of computation to pipelined datapaths
  - Makes virtual architecture possible
  - Simplifies compilation and programming

Pipelined Datapaths

```
for (i=0; i<maxIn; i++)
{
   y[i]=0;
   for (j=0; j<Taps; j++)
   {
     y[i] += x[i+j]*w[j];
   }
}
```



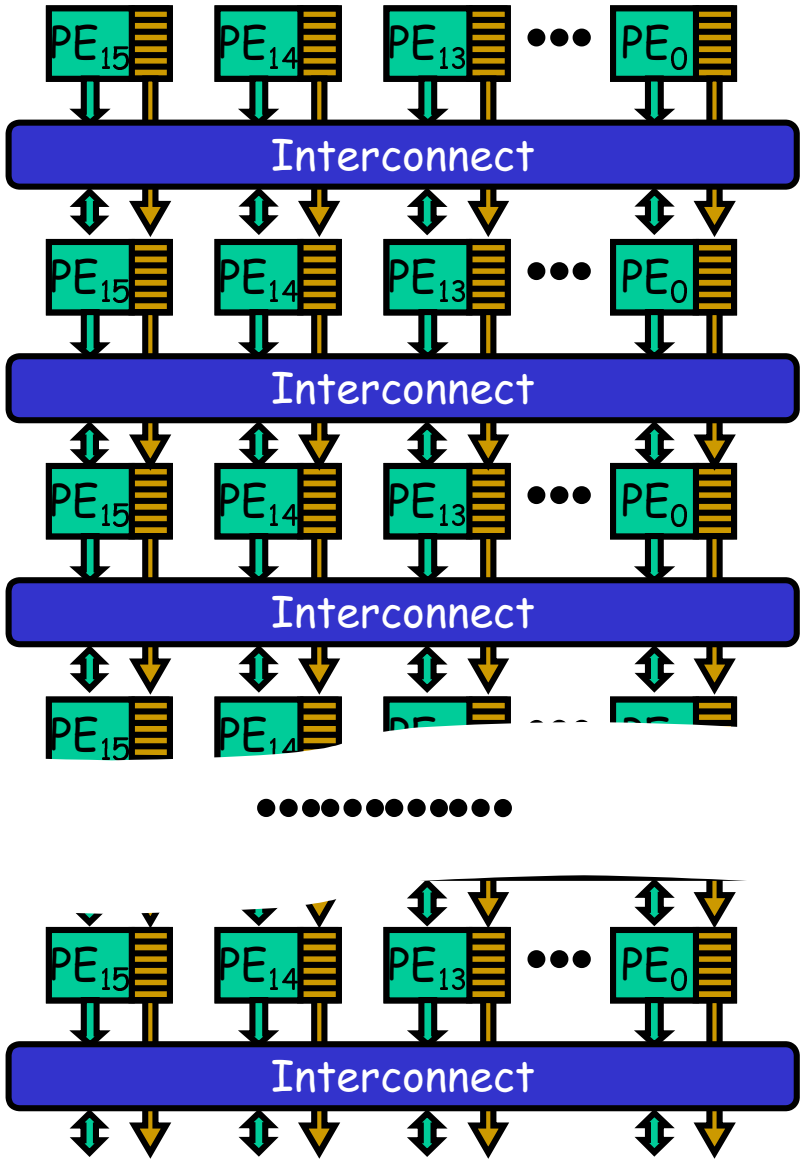
Lots of apps fit:

- DSP
- Image Processing
- Cryptography

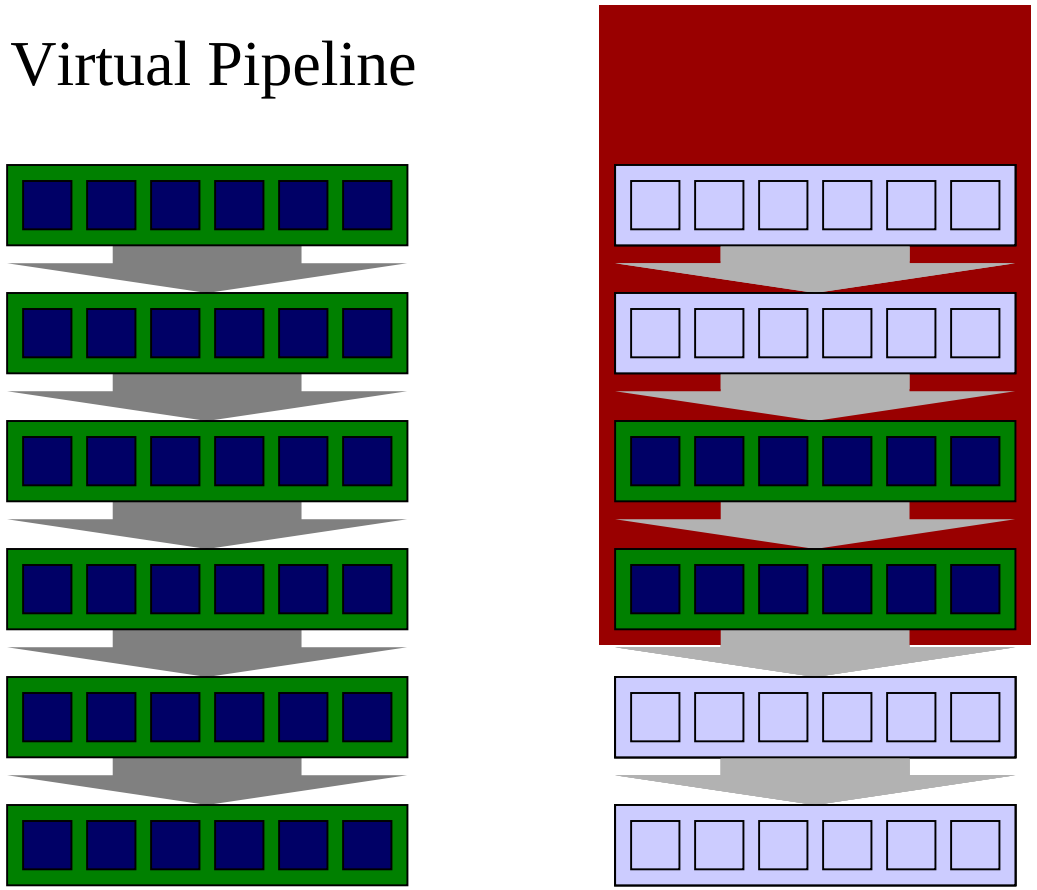
PipeRench Fabric

A programmable, pipelined data path containing:

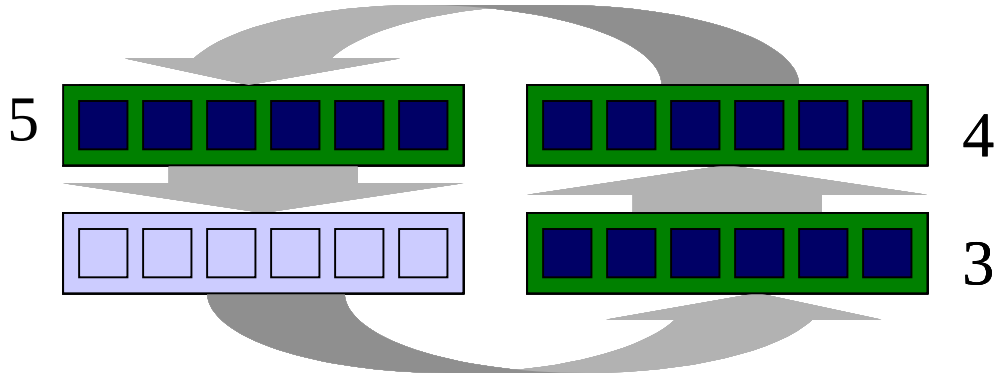
Processing elements
Local interconnect
Pass Registers
Unbounded Depth



Pipeline Virtualization

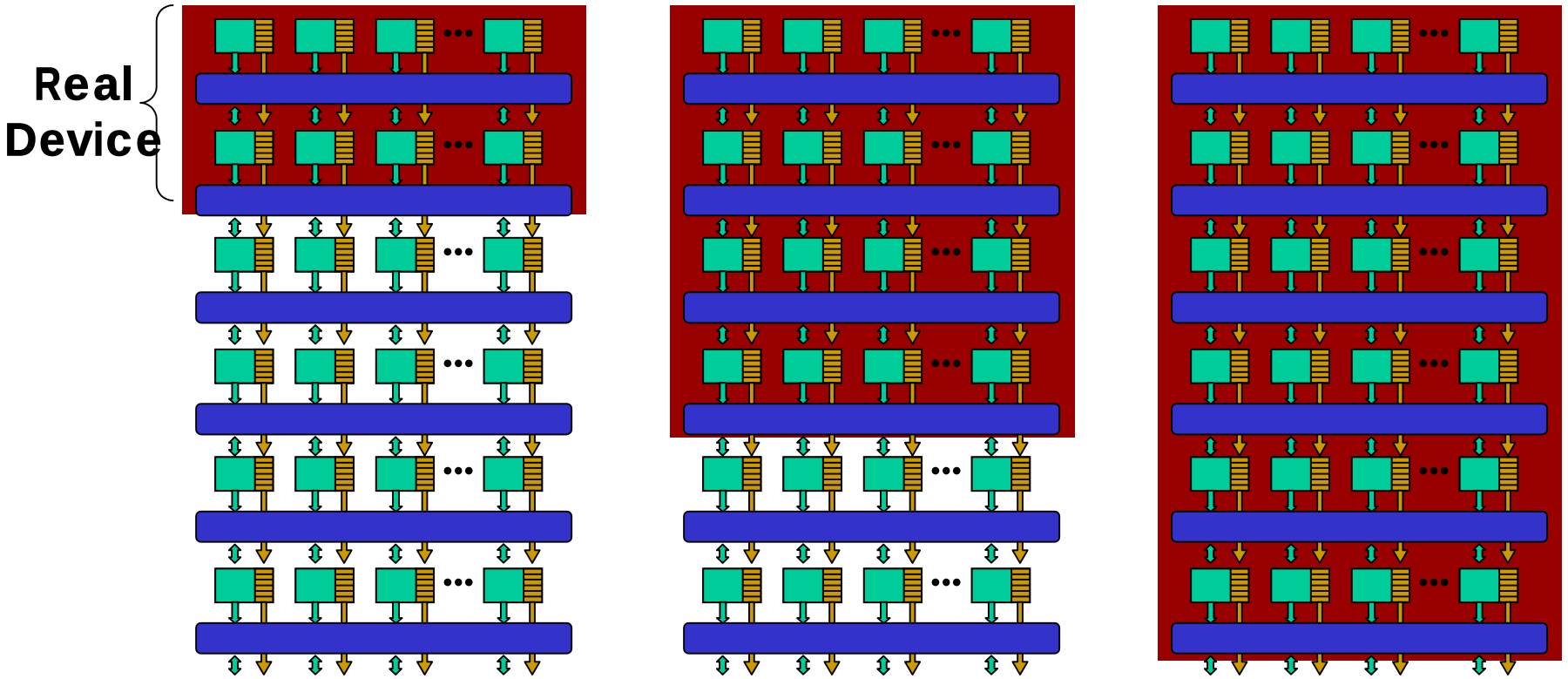


Pipeline Virtualization



Since stripes are connected in a ring, data can always pass between adjacent virtual stripes in the physical fabric.

Performance Scaling



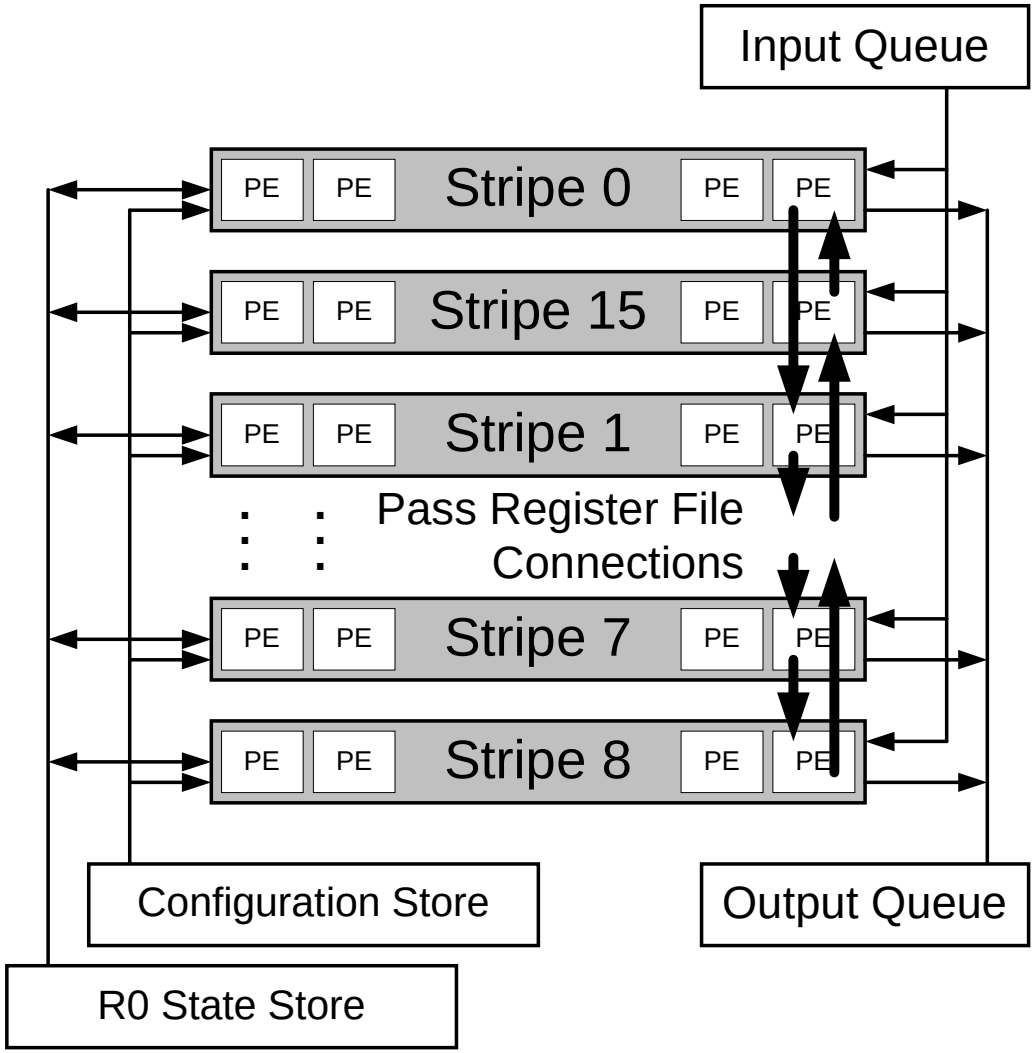
2 Stripes
2 Outputs / 6 Cycles

4 Stripes
4 Outputs / 6 Cycles

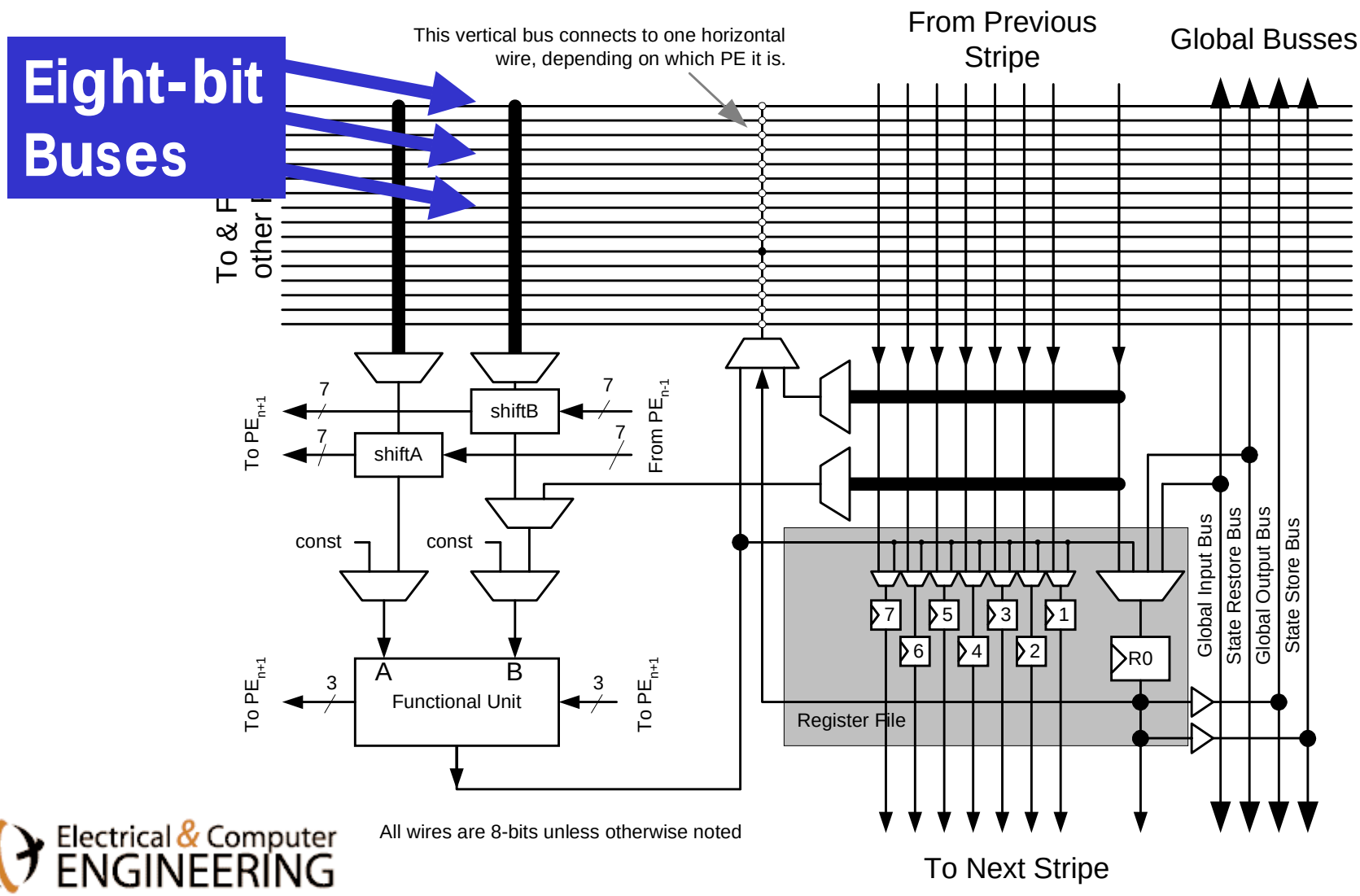
6 Stripes
6 Outputs / 6 Cycles

Chip Structure

- Ring Structure
  - Interleaved
- Global Buses
  - Inputs
  - Outputs
  - Configuration
  - State Storage
- 16 Stripes
  - 16 PEs each

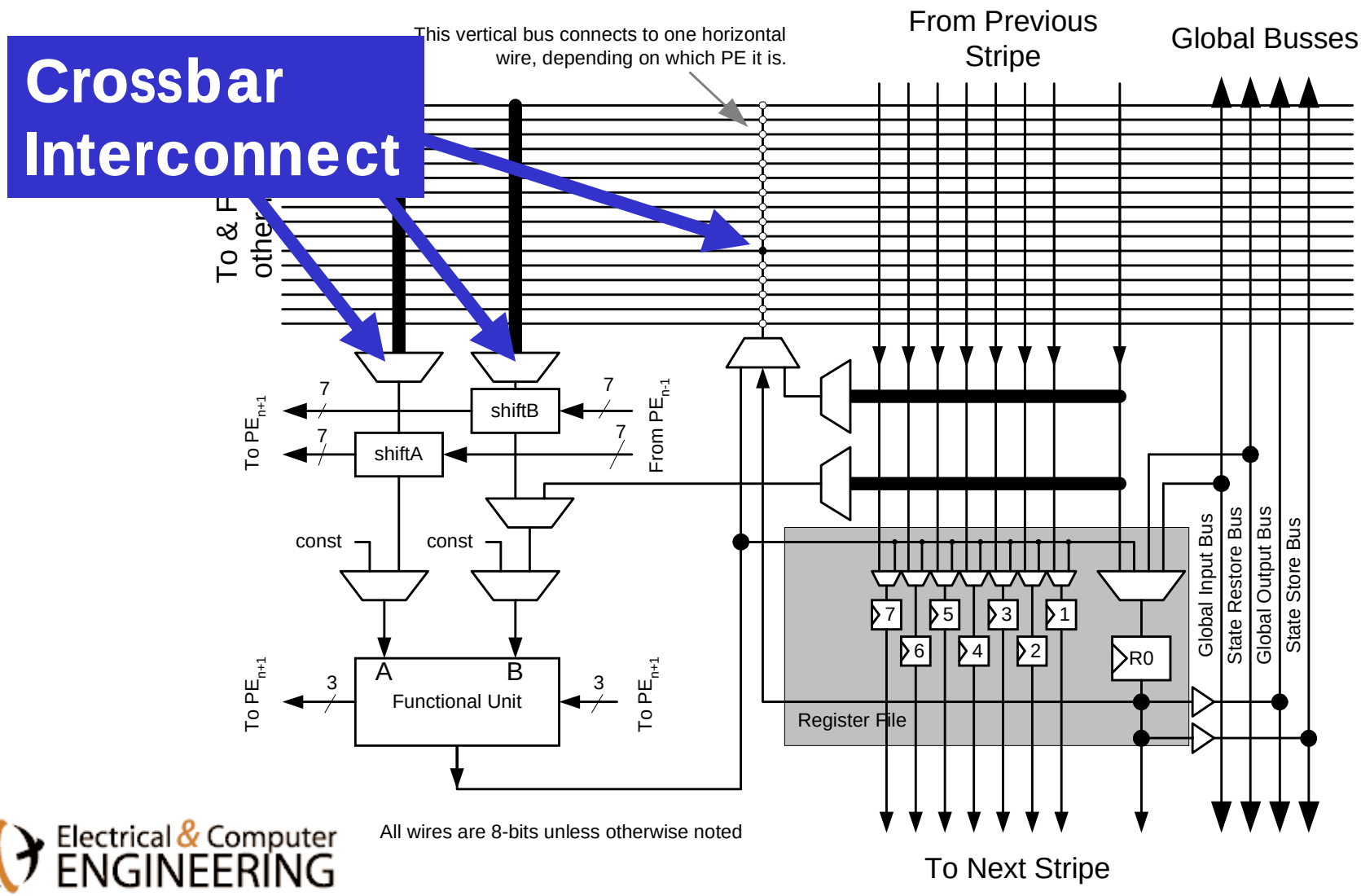


PE Architecture



Electrical & Computer ENGINEERING

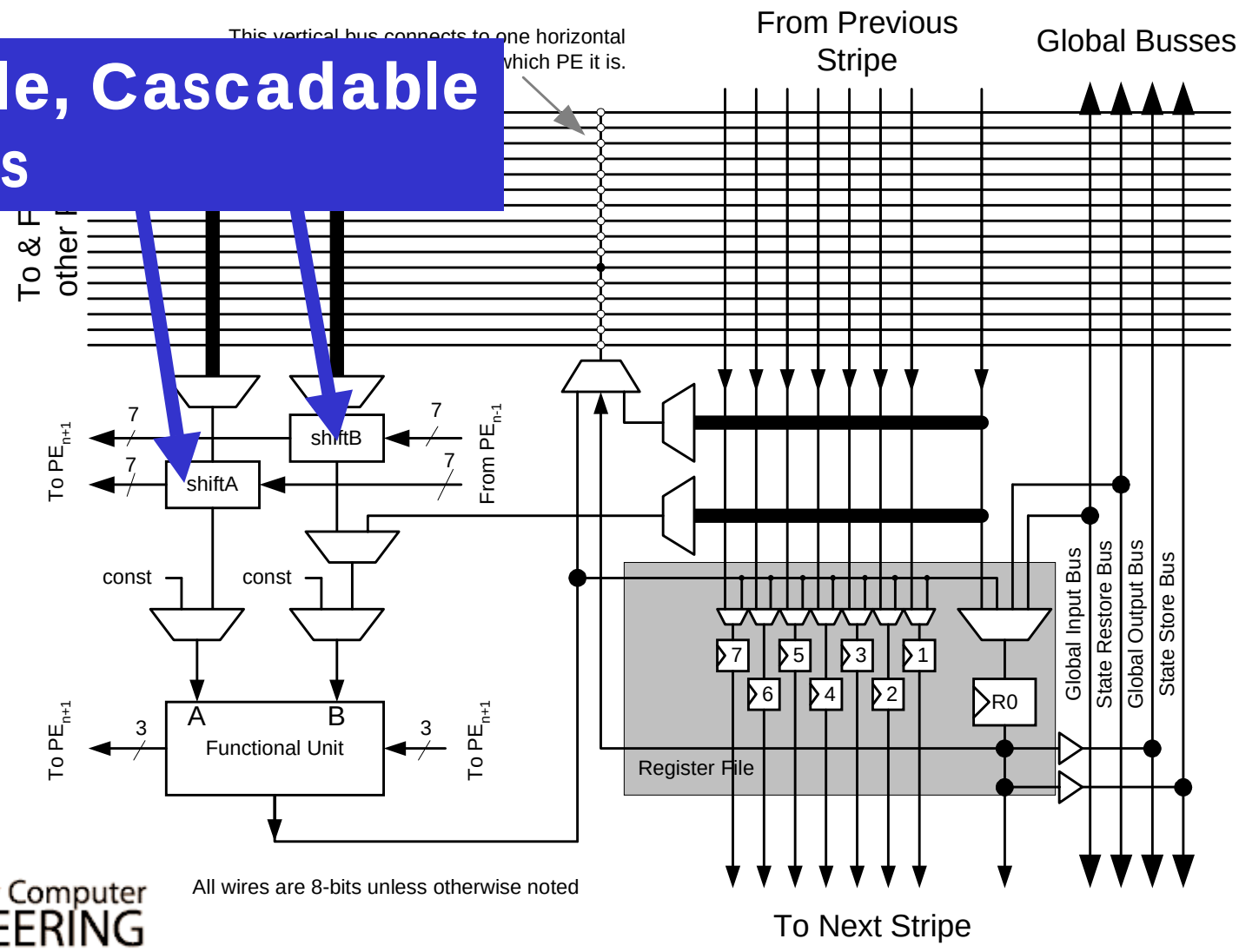
PE Architecture



Electrical & Computer ENGINEERING

PE Architecture

Flexible, Cascadable Shifters

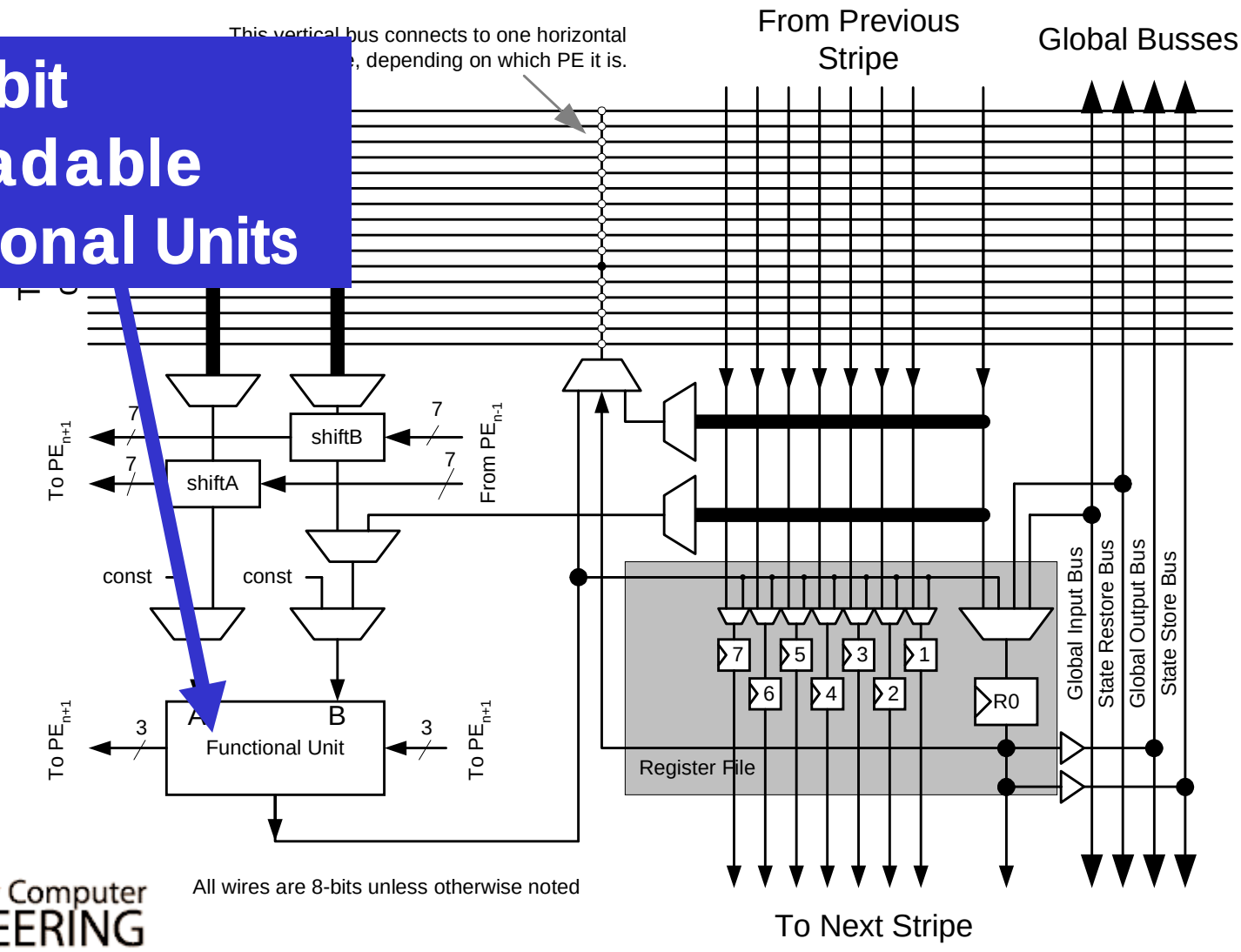


All wires are 8-bits unless otherwise noted

Electrical & Computer ENGINEERING

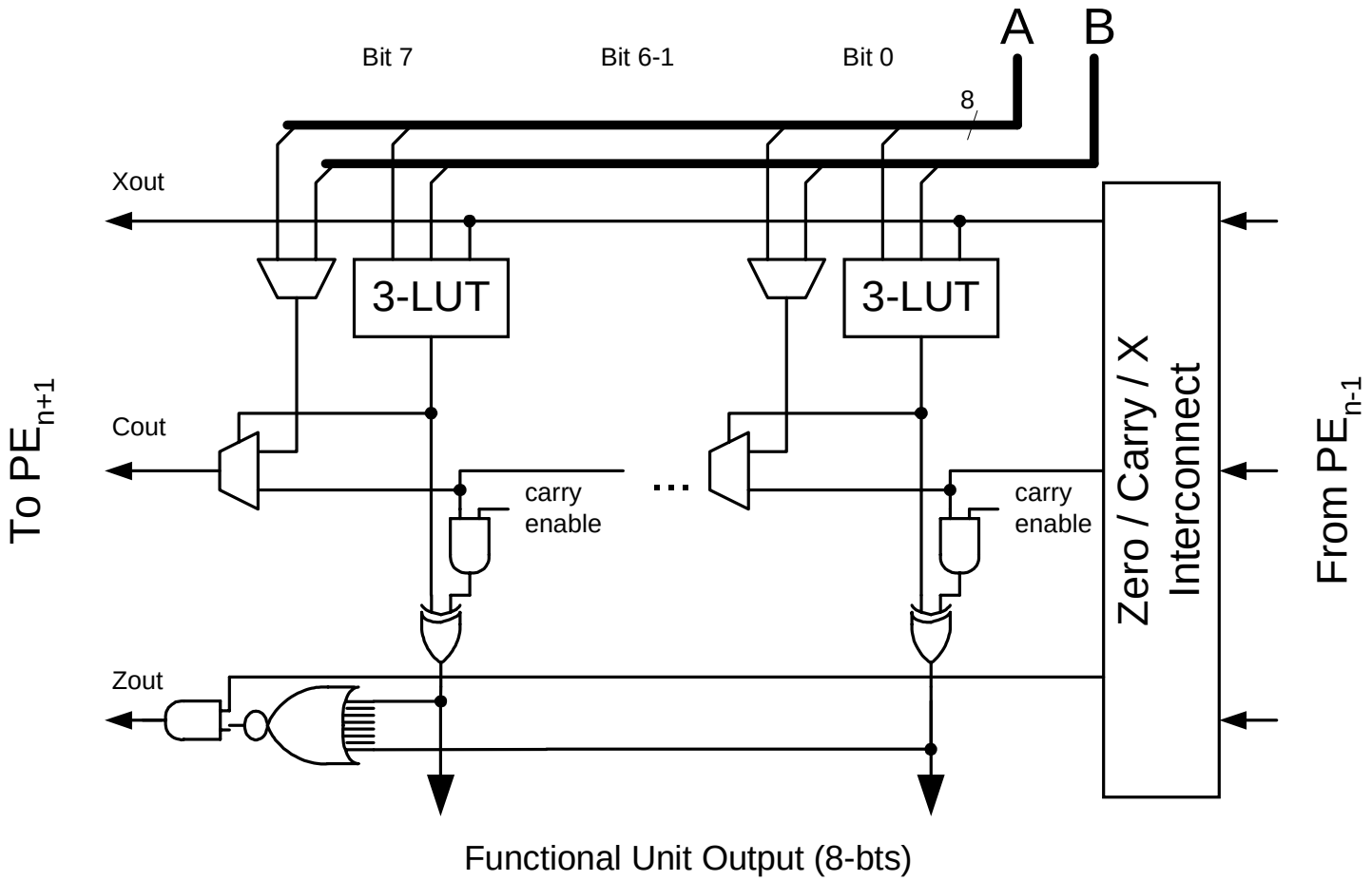
PE Architecture

Eight-bit Cascadable Functional Units

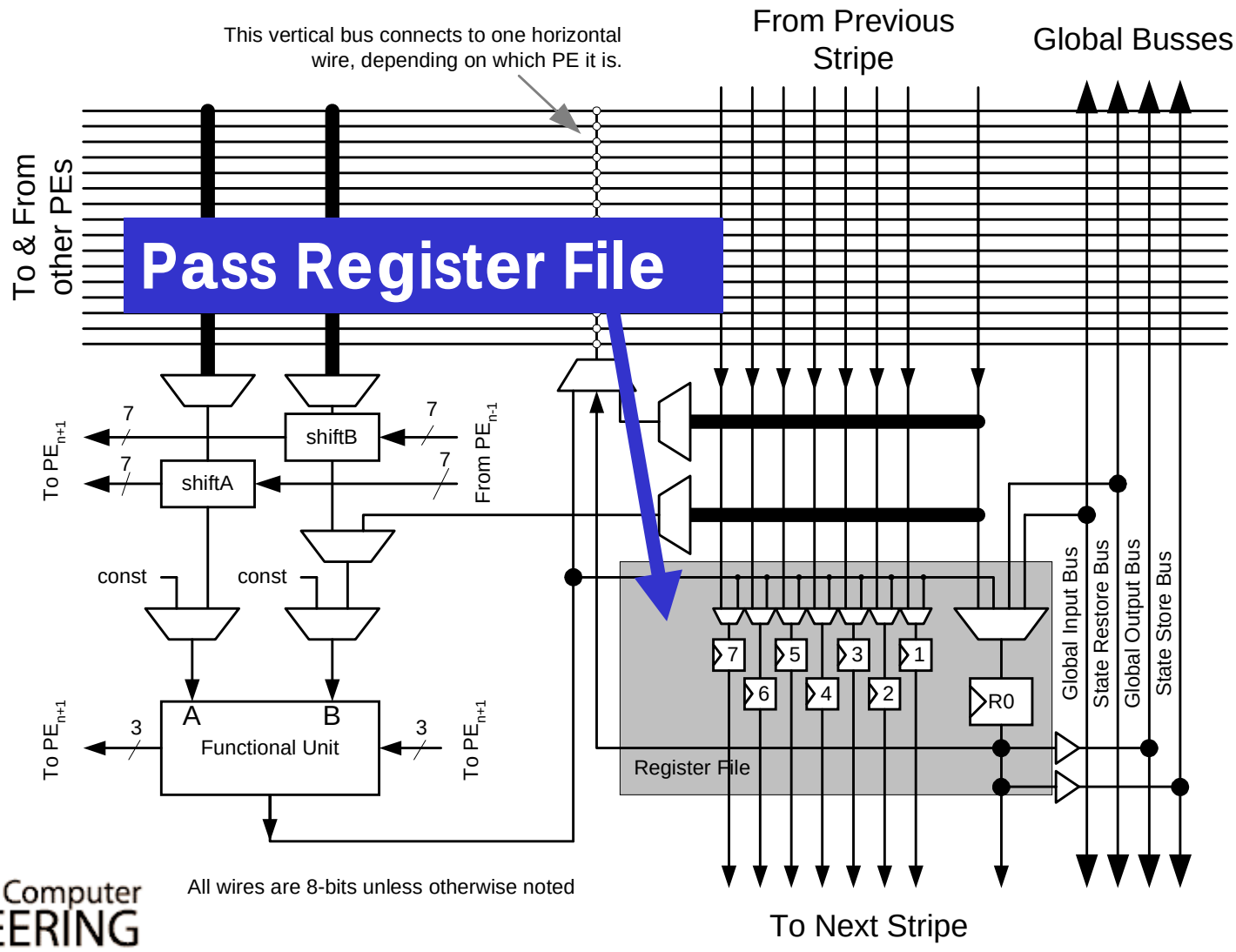


Electrical & Computer ENGINEERING

Functional Unit Architecture



PE Architecture

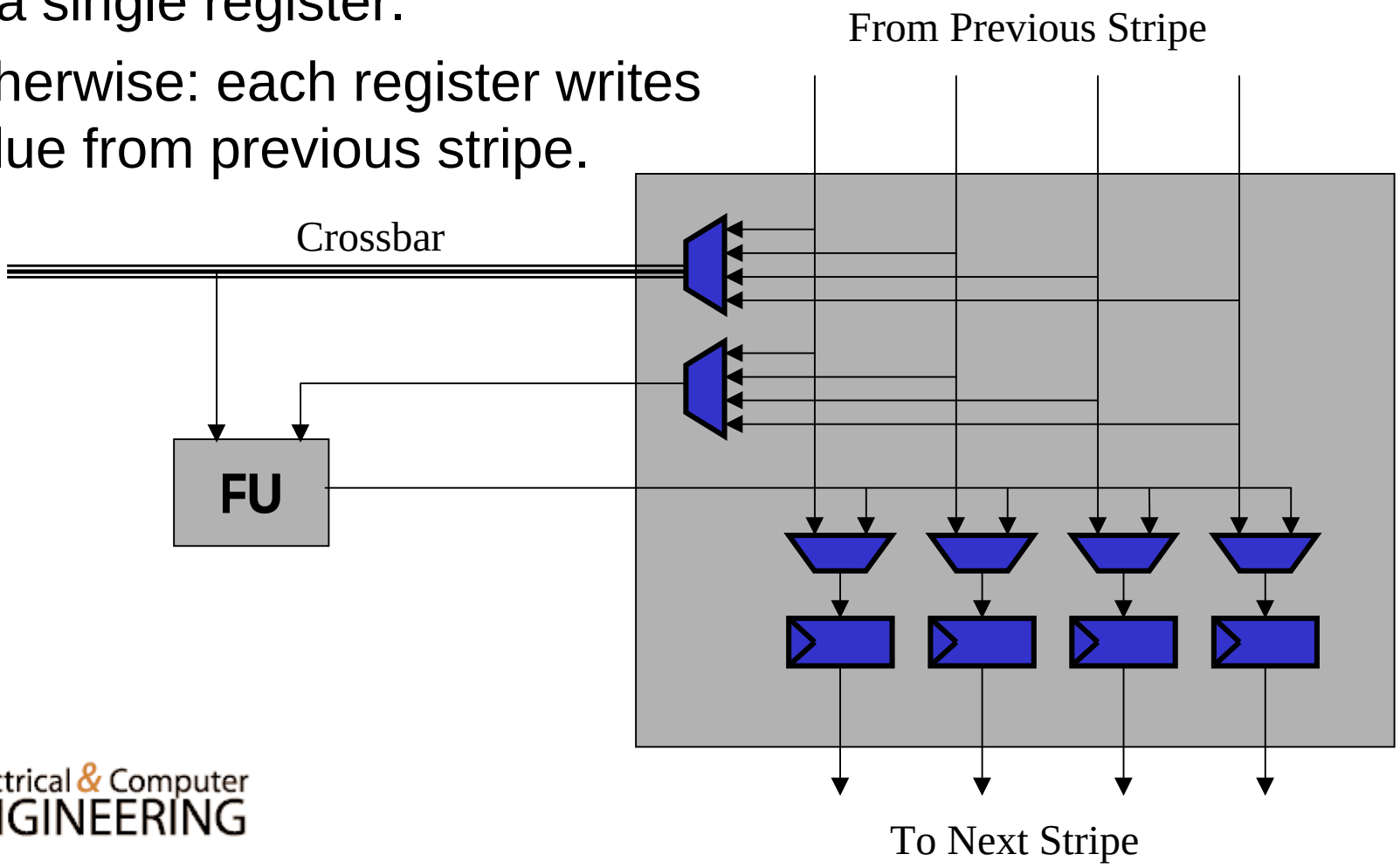


All wires are 8-bits unless otherwise noted

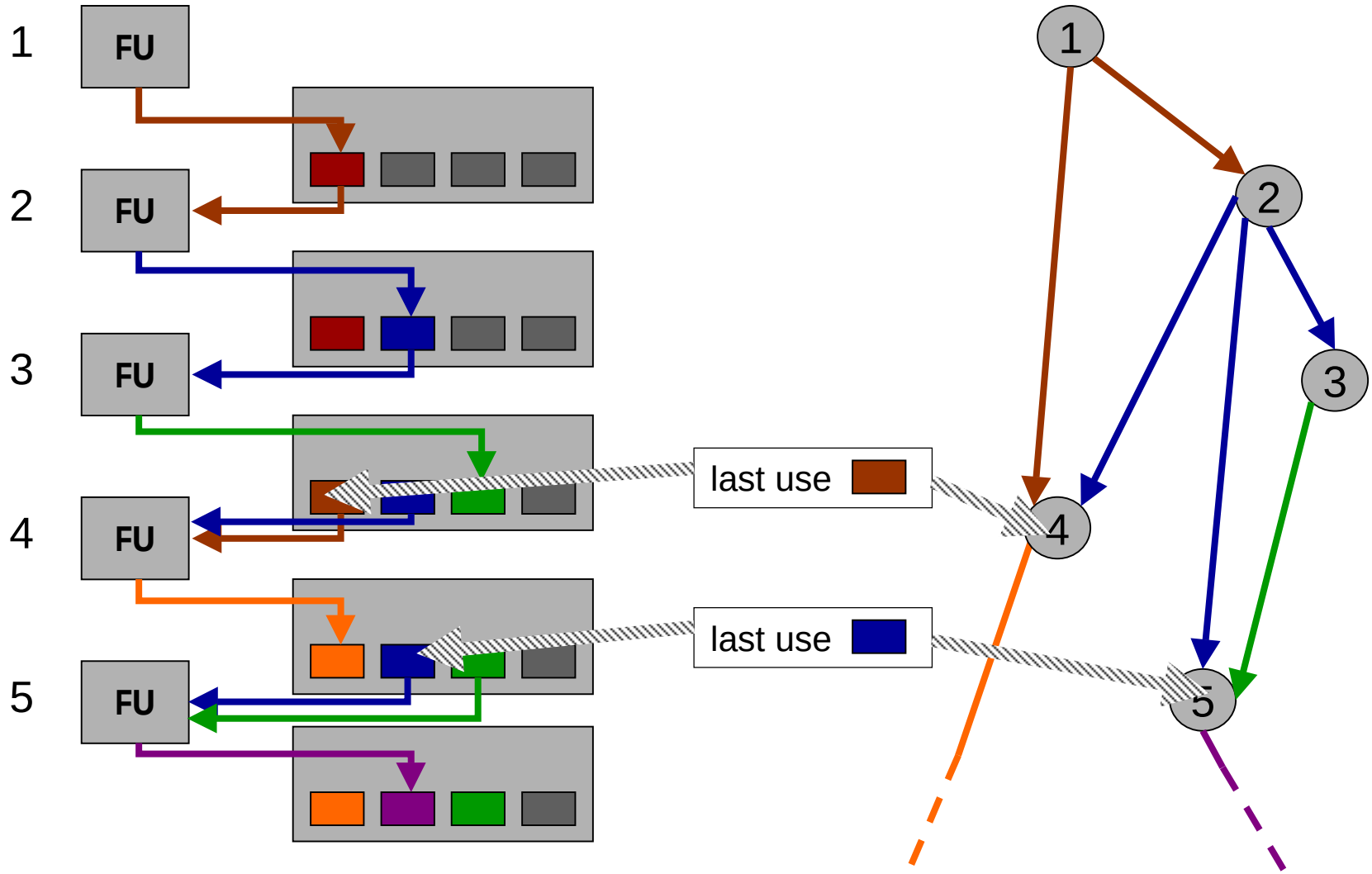
Electrical & Computer ENGINEERING

Pass Register File

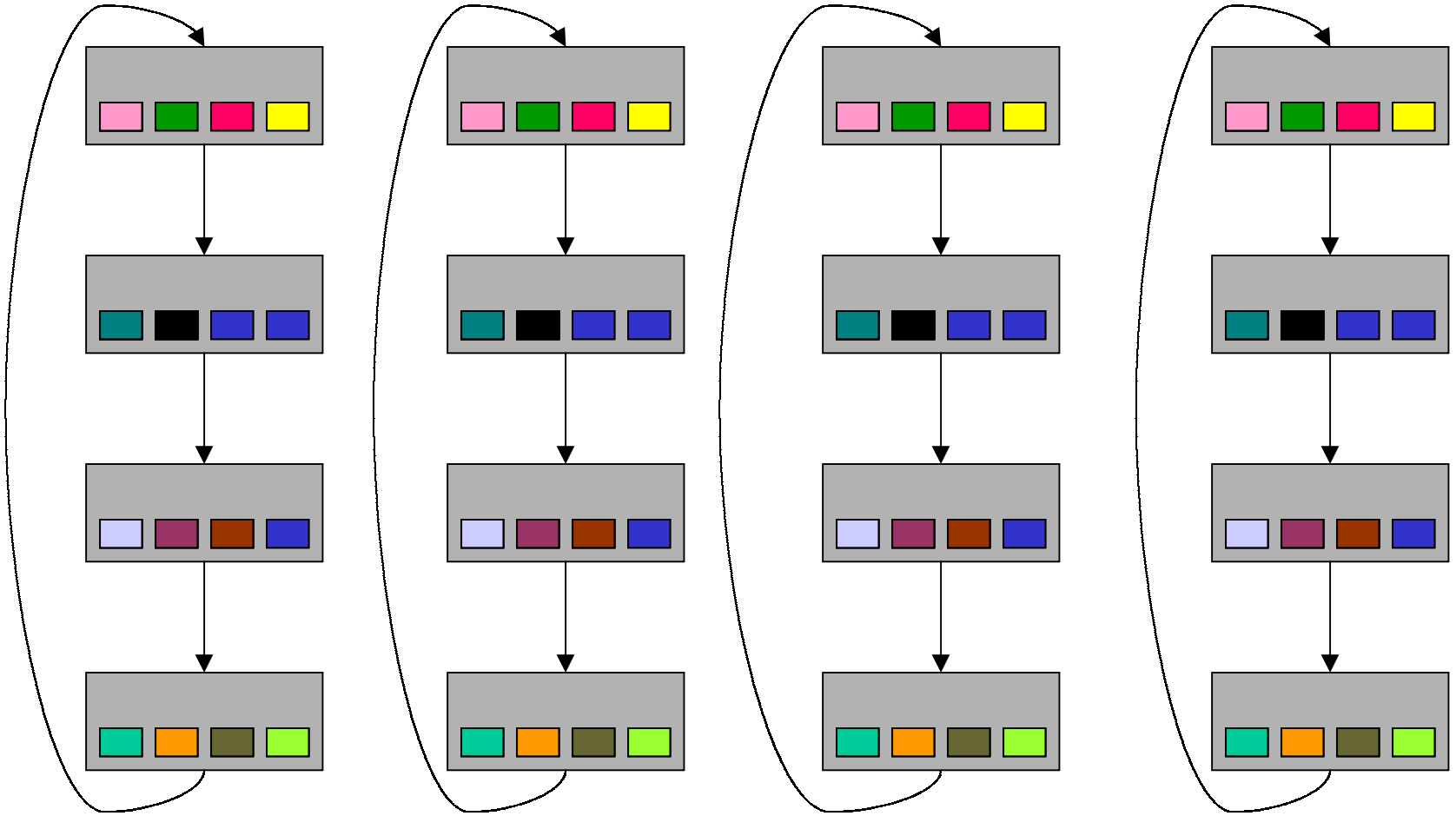
- Two values can be read in each stripe.
- PE can write one new value to a single register.
- Otherwise: each register writes value from previous stripe.



Pass Register File Operation



Pass Register Problem

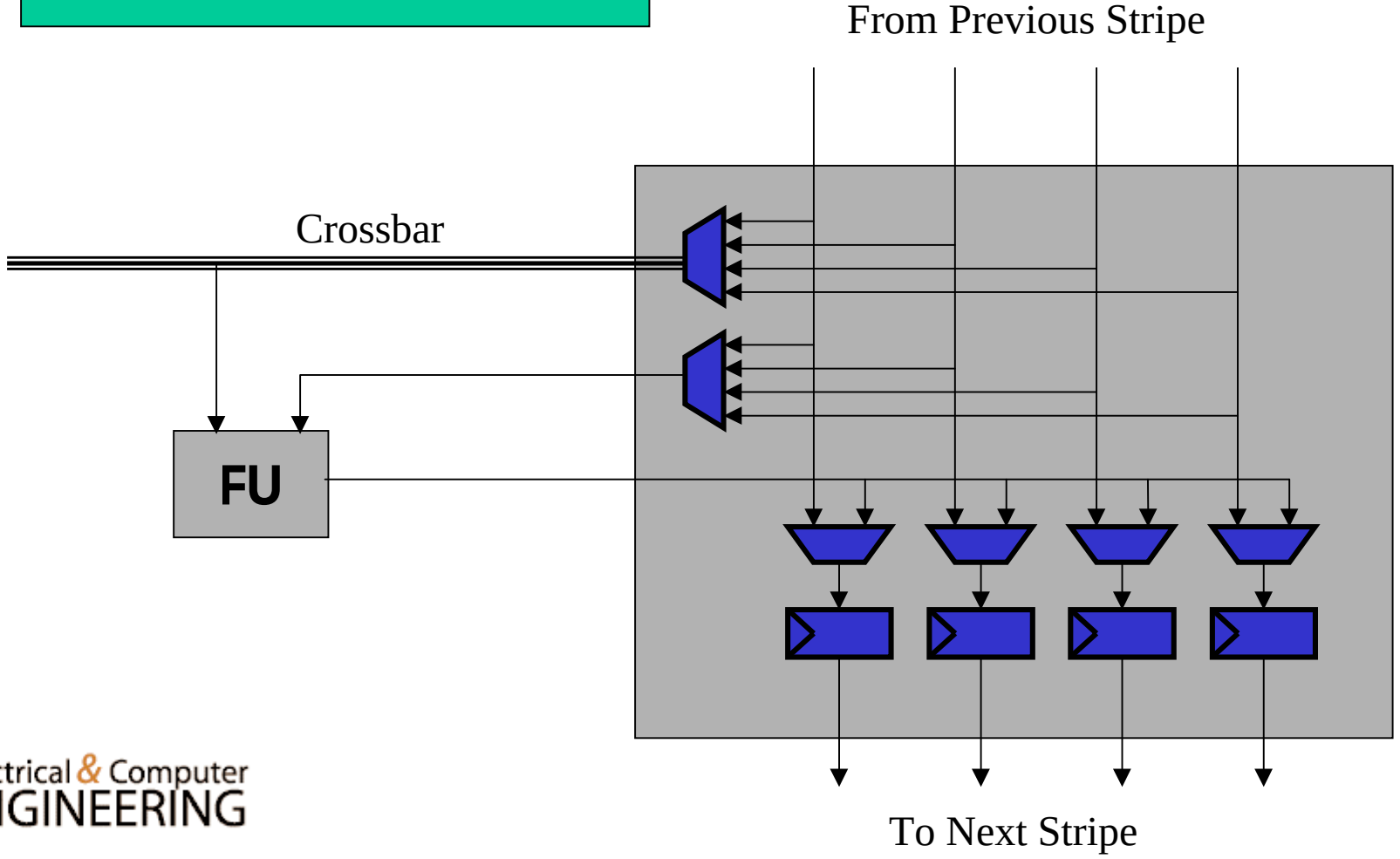


Old register values cycle through fabric endlessly.
Extra switching consumes power.

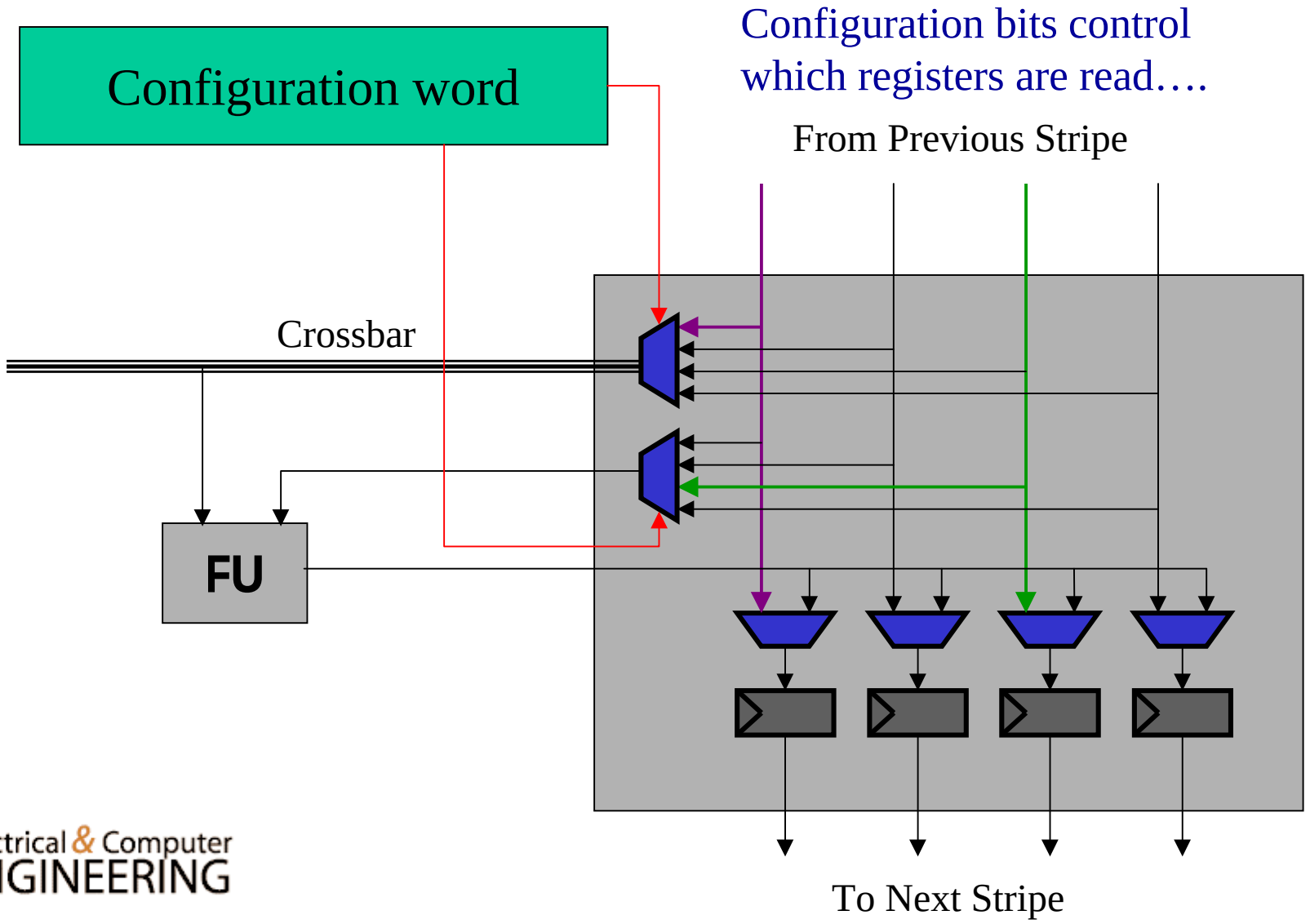
Solution: Register Kill

Configuration word

Configuration bits control which registers are read….



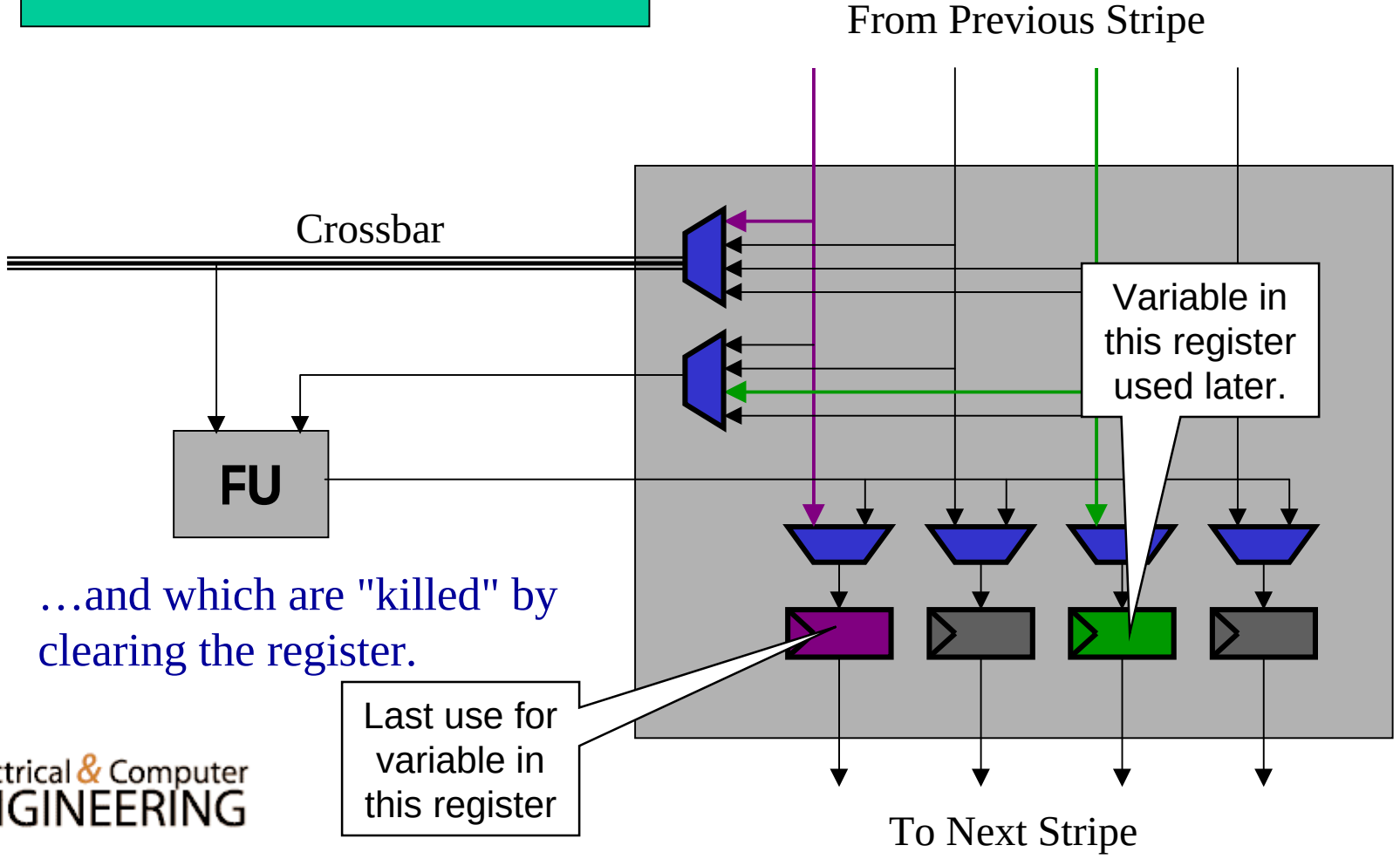
Solution: Register Kill



Solution: Register Kill

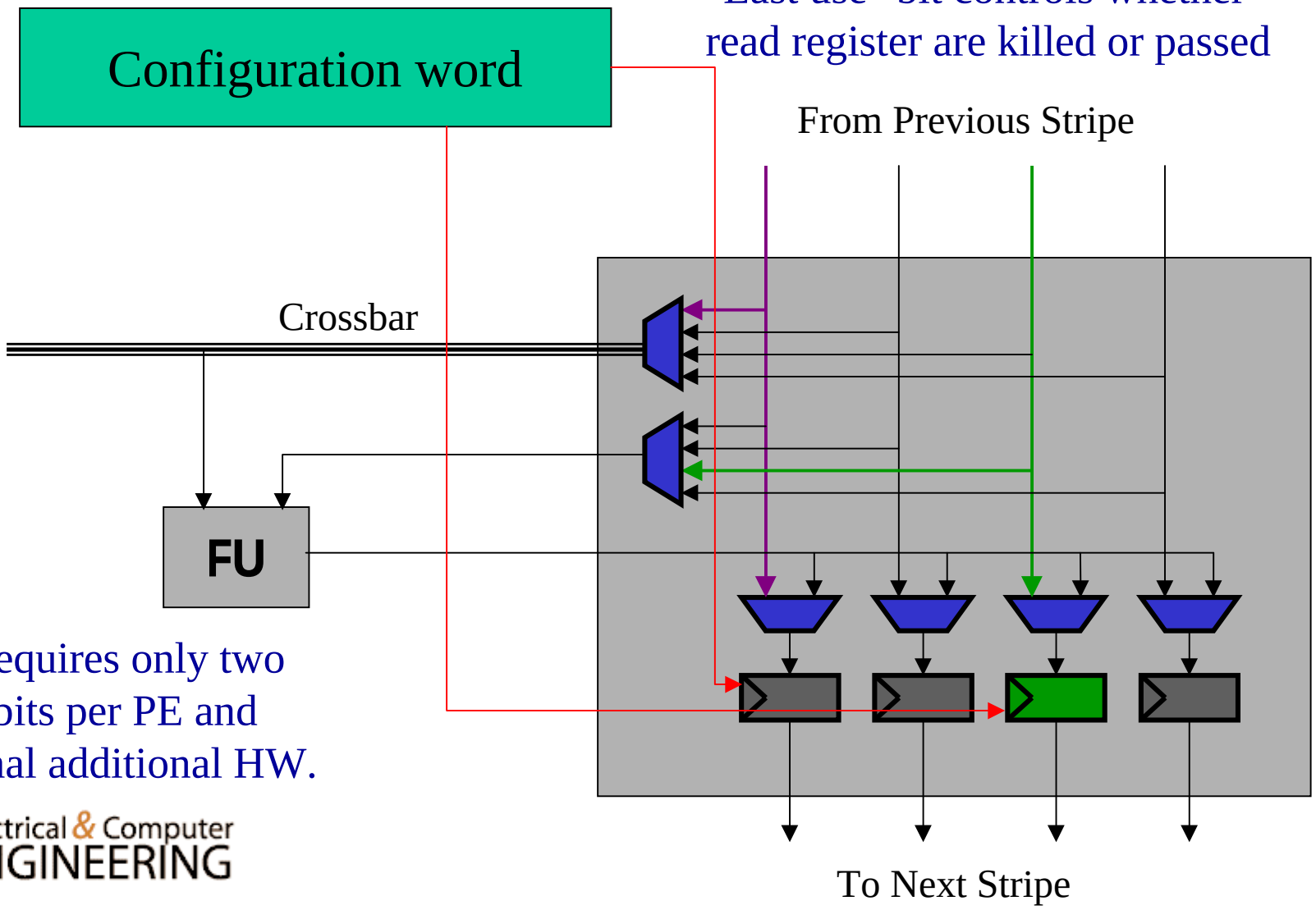
Configuration word

…they can also control which registers are passed…



…and which are "killed" by clearing the register.

Solution: Register Kill



This requires only two more bits per PE and minimal additional HW.

Implemented Hardware Design

- Industrial Partner: ST Microelectronics
- Process Technology: 0.18 micron, 6 metal

  3.65 million transistors

  49 sq. mm die

  120 MHz fabric operation

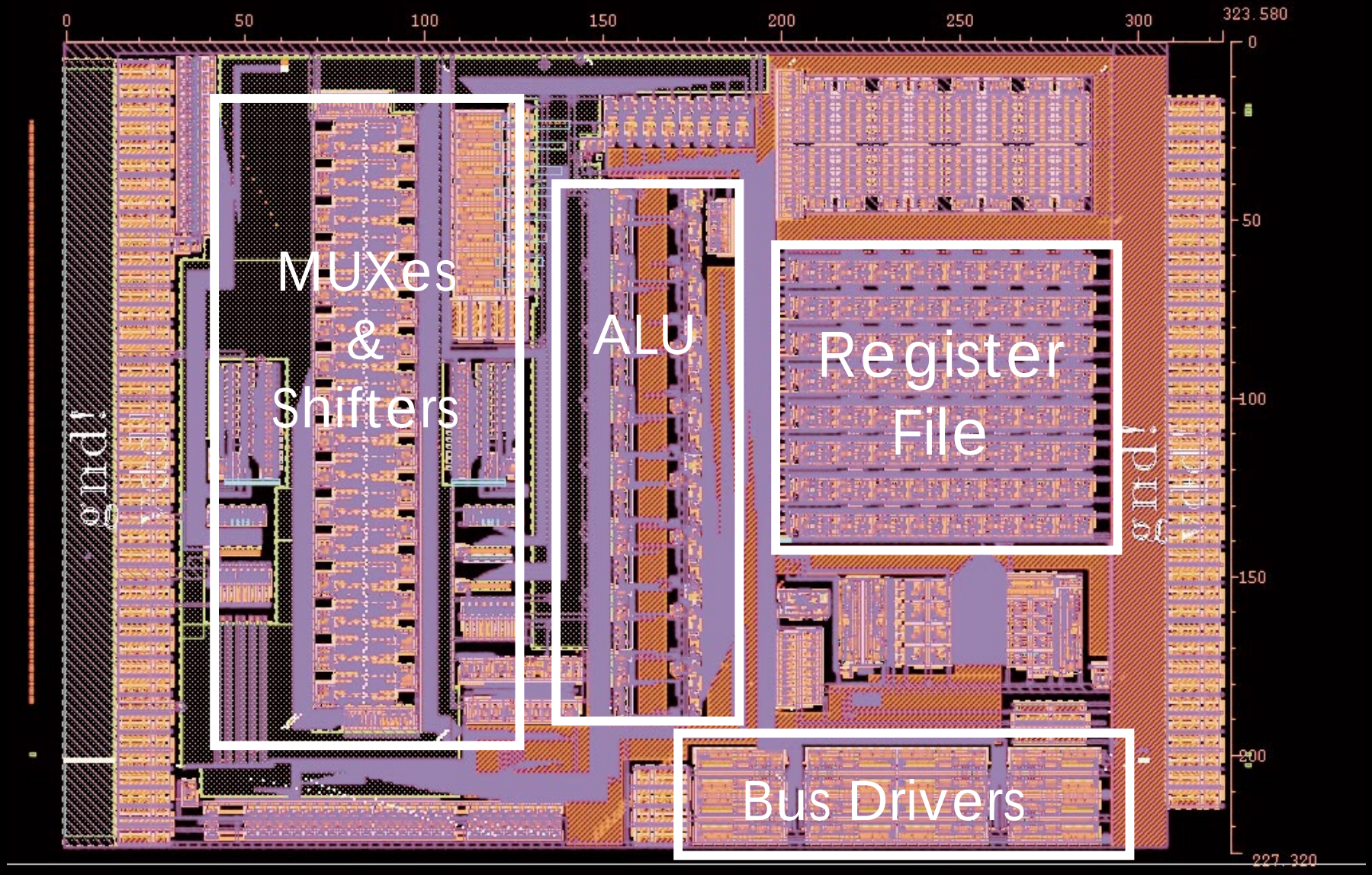
  60 MHz I/O frequency

  < 3W power

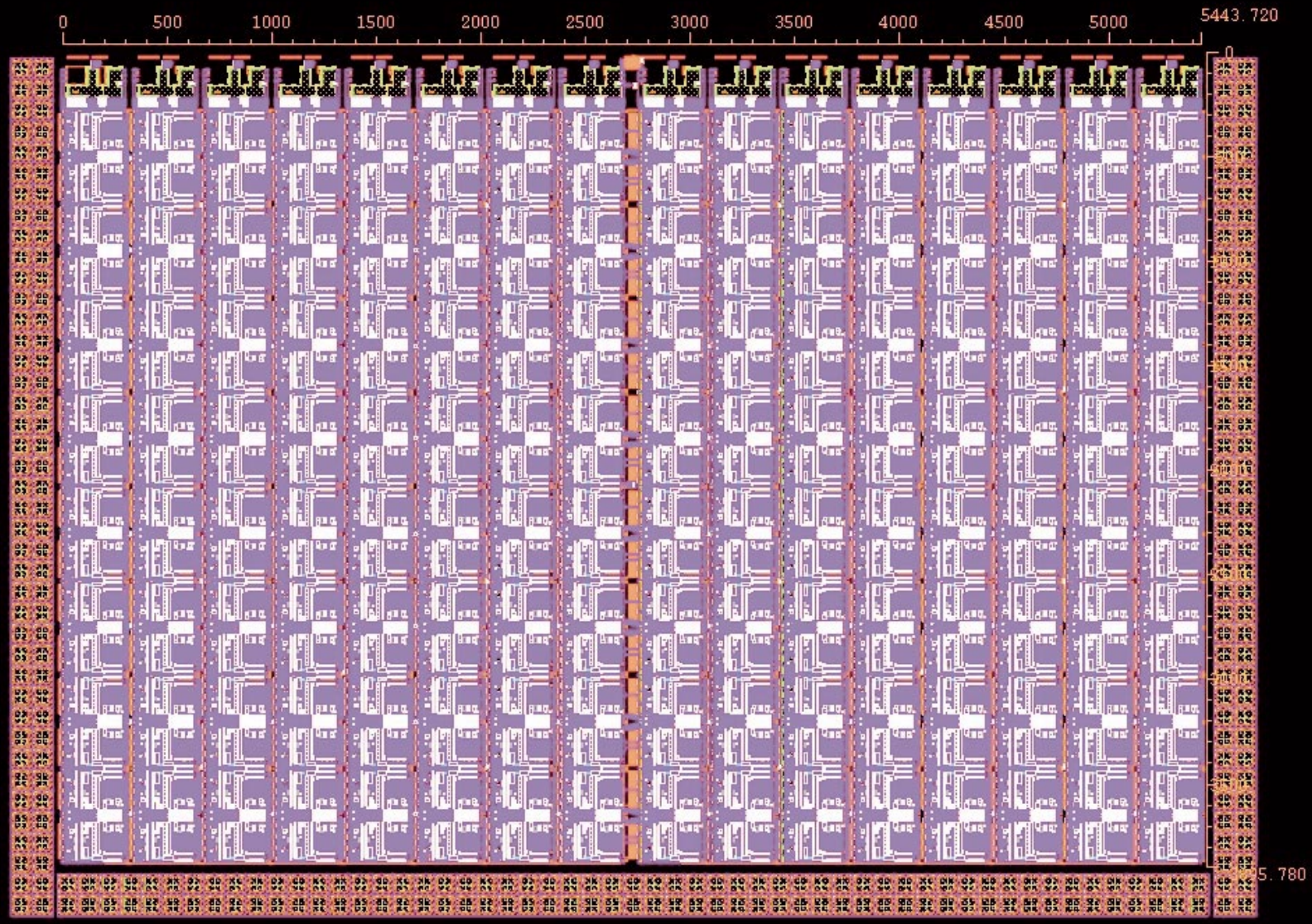
  Reconfigure entire fabric in 133ns.

  Switch applications in 8ns.

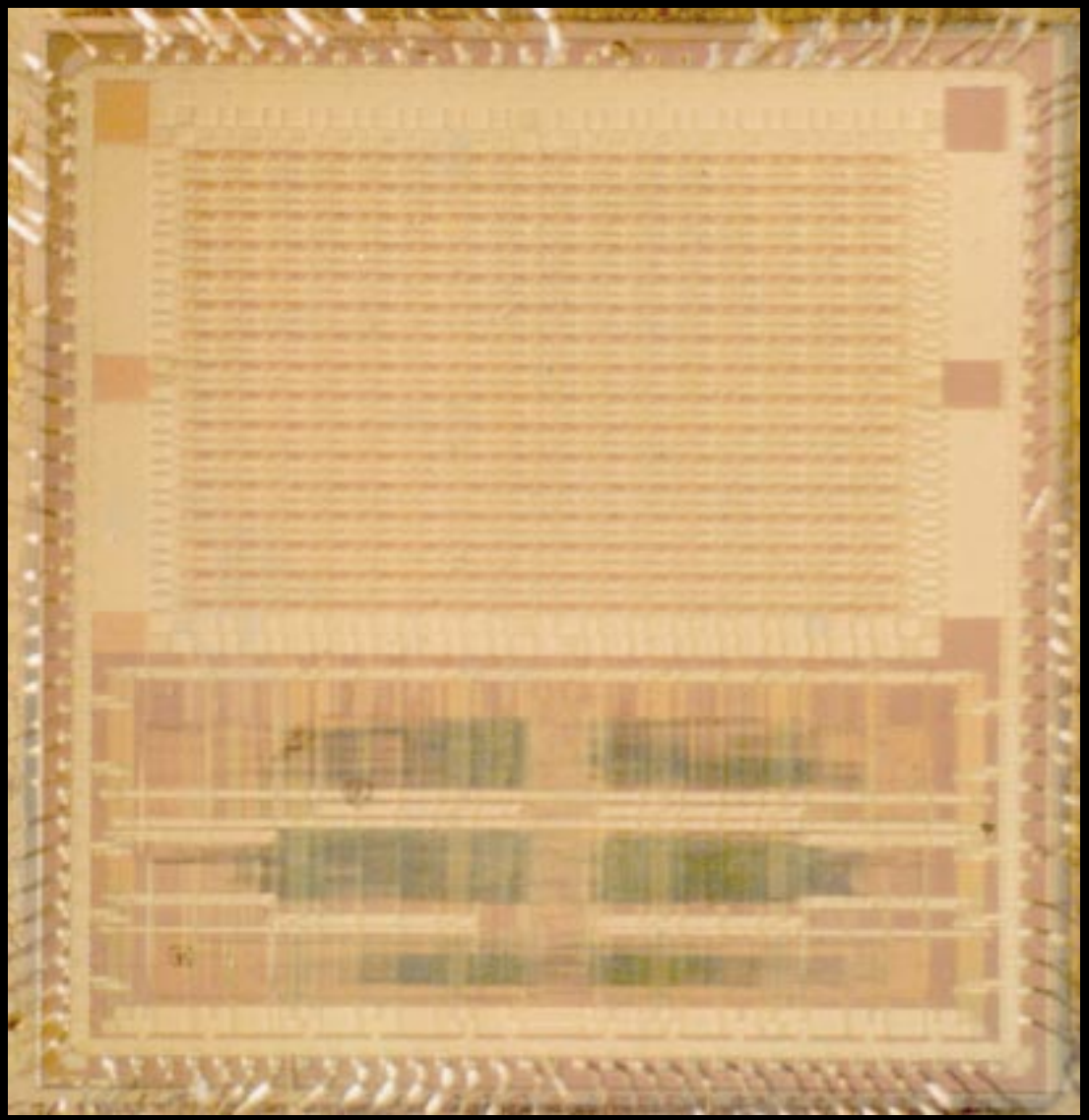
Tile Layout


0
50
100
150
200
250
300
323.580
0
50
100
150
200
227.320
MUXes
&
Shifters
ALU
Register
File
Bus Drivers

Fabric Layout



Chip Die Shot



Performance on Filtering

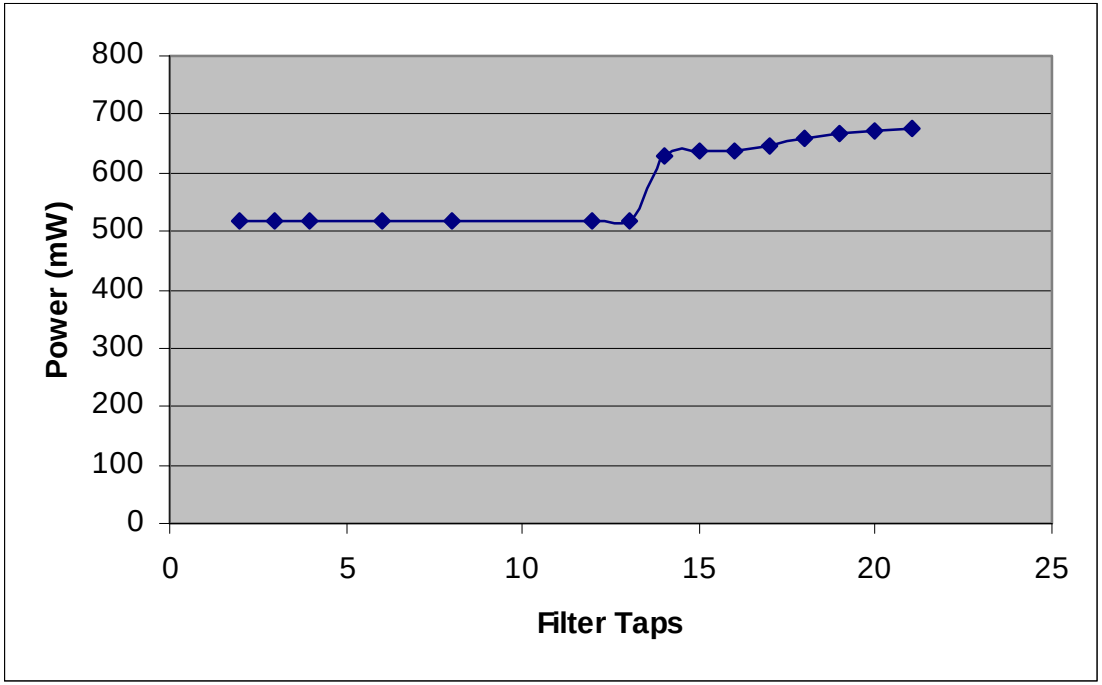
- 40 Tap 16-bit FIR Filter
  - 41.8 MSPS
- Comparable to high-end DSPs
  - Much lower clock rate
  - Without a full multiplier

    (taps are compiled into hardware)

Performance on Encryption

- IDEA Encryption: 450 Mbps
  - Key is compiled into hardware
  - Compilation (including P&R) takes less than one minute
- Comparison:
  - 800 MHz Pentium III Xenon: 75.4 Mbps

Power Consumption – FIR Filter



- Before 14 taps, near constant power
- At 14 taps, virtualization causes step

Conclusions

- A practical virtual machine for pipelined programmable datapaths is possible.
- Virtual hardware ⇒ physical hardware:
  - Completely self-managed on chip at run-time.
  - Enabled by fast incremental reconfiguration.
- Virtual architecture allows:
  - Easier compilation
  - Forward compatibility / Scalability
- Implemented chip has high performance and low power requirements